CLAIMS

What is claimed is:

1. A method of handling a data transfer in a network interface controller (NIC), the method comprising the steps of:

a) receiving the data transfer wherein the data transfer is denoted as one of a first type and a second type;

b) calculating a cyclical redundancy check (CRC) for the data transfer, wherein the CRC is one of valid and invalid; and

c) conducting one of:

1) dropping the data transfer and not confirming reception;

2) placing the data transfer to a reassembly buffer of the NIC; and

3) placing the data transfer to an internal buffer of the NIC for direct data placement to a destination buffer.

2. The method of claim 1, wherein step c), 2) is conducted in the case that the data transfer is of the first type.

3. The method of claim 1, further comprising the step of determining whether the data transfer includes a single or multiple direct data placement (DDP) segments.

4. The method of claim 3, wherein step c), 3) is conducted in the case that the data transfer includes multiple DDP segments and all DDP segments have a valid CRC that is fully contained in a TCP segment.

5. The method of claim 3, wherein step c), 1) is conducted in the case that the data transfer includes multiple DDP segments, a first DDP segment has an invalid CRC, and a DDP header of the first DDP segment is referred by an MPA length associated with a previous DDP segment.

6. The method of claim 5, wherein, in the case that the data transfer includes multiple DDP segments, a first DDP segment has an invalid CRC, and the DDP header of the first DDP segment is not referred by the MPA length associated with the previous DDP segment:

step c), 1) is conducted in the case that the DDP header is referred by an MPA marker; and

step c), 2) is conducted in the case that the DDP header is not referred by the MPA marker.

7. The method of claim 3, wherein step c), 1) is conducted in the case that the data transfer includes multiple DDP segments and a last DDP segment extends outside of the TCP segment boundary;

and step c), 2) is conducted in the case that the data transfer includes multiple DDP segments and a last DDP segment does not extend outside of the TCP segment boundary.

8. The method of claim 2, wherein step c), 2) is conducted in the case that the data transfer includes a single DDP segment and an MPA length associated with the single DDP segment is greater than a transmission control protocol (TCP) segment length of the data transfer.

9. The method of claim 2, wherein step c), 3) is conducted in the case that the data transfer includes a single DDP segment that has: an MPA length associated therewith that equals a TCP segment length and a valid CRC.

10. The method of claim 2, wherein step c), 1) is conducted in the case that the data transfer includes a single DDP segment that has: an MPA length associated therewith that equals a TCP segment length, an invalid CRC and a DDP header that is referred by an MPA length associated with a previous DDP segment.

11. The method of claim 2, wherein in the case that the data transfer includes a single DDP segment that has: an MPA length associated therewith that equals a TCP segment length, an invalid CRC and a DDP header that is not referred by an MPA length associated with a previous DDP segment:

step c), 1) is conducted in the case that the DDP header is referred by an MPA marker; and

step c), 2) is conducted in the case that the DDP header is not referred by an MPA marker.

12. The method of claim 1, further comprising the step of setting the data transfer type to the first type when step c), 2) is conducted.

13. The method of claim 1, wherein in the case that step c), 3) is conducted on an out-of-order data transfer, the method further comprises the steps of:

clearing TCP hole information created by the out-of-order data transfer in a connection context; and

stopping receipt reporting for the out-of-order data transfer.

14. The method of claim 1, wherein the data transfer includes DDP segments, and the calculating step includes calculating a CRC for all DDP segments of the data transfer together.

15. The method of claim 14, wherein the data transfer does not contain an MPA marker.

16. The method of claim 14, further comprising the steps of:

storing a number of retransmission attempts for each data transfer including an error; and

storing a largest sequence number.

17. The method of claim 16, wherein in the case that CRC is invalid for the data transfer, which indicates the data transfer is a newly received error-including data transfer:

step c), 2) is conducted on the newly received error-including data transfer in the case that the number of retransmission attempts exceeds a maximum retransmission attempt number for

that data transfer, and

step c), 1) is conducted on the newly received error-including data transfer in the case that the number of retransmission attempts does not exceed a maximum retransmission attempt number for that data transfer; and

wherein in the case that step c), 1) is conducted, the method further comprises the steps of:

increasing the number of retransmission attempts for the newly received error-including data transfer by one; and

updating the largest sequence number to carry the largest sequence number among at least one previously received error-including data transfer and the newly received error-including data transfer.

18. The method of claim 16, wherein in the case that CRC is valid for an in-order data transfer:

a) in the case that a sequence number of the in-order data transfer is greater than the stored largest sequence number, the number of retransmission attempts is reset and step c), 3) is conducted; and

b) in the case that the sequence number of the in-order data transfer is not greater than the stored largest sequence number, step c), 3) is conducted.

19. A network interface controller (NIC) for handling a data transfer, the NIC comprising:

first storage means for storing the data transfer for reassembly;

second storage means for storing the data transfer for direct data placement to a destination buffer;

means for receiving the data transfer wherein the data transfer is denoted as one of a first type and a second type;

means for calculating a cyclical redundancy check (CRC) for the data transfer, wherein the CRC is one of valid and invalid; and

means for conducting one of:

1) dropping the data transfer and not confirming reception;

2) placing the data transfer to a reassembly buffer of the NIC; and

3) placing the data transfer to an internal buffer of the NIC for direct data placement to a destination buffer.

20. The NIC of claim 19, wherein the conducting means conducts c), 2) in the case that the data transfer is of the first type.

21. The NIC of claim 19, further comprising means for determining whether the data transfer includes a single or multiple direct data placement (DDP) segments.

22. The NIC of claim 21, wherein the conducting means conducts c), 3) in the case that the data transfer includes multiple DDP segments and all DDP segments have a valid CRC that are fully contained in TCP segment.

23. The NIC of claim 21, wherein the conducting means conducts c), 1) in the case that the data transfer includes multiple DDP segments, a first DDP segment has an invalid CRC, and a DDP header of the first DDP segment is referred by an MPA length associated with a previous DDP segment.

24. The NIC of claim 21, wherein in the case that the data transfer includes multiple DDP segments, a first DDP segment has an invalid CRC, and a DDP header of the first DDP segment is not referred by an MPA length associated with a previous DDP segment:

step c), 1) is conducted in the case that the DDP header is referred by an MPA marker; and

step c), 2) is conducted in the case that the DDP header is not referred by the MPA marker.

25. The NIC of claim 21, wherein the conducting means conducts c), 1) in the case that the data transfer includes multiple DDP segments and a last DDP segment extends outside of the TCP segment boundary;

and conducts c), 2) in the case that the data transfer includes multiple DDP segments and a last DDP segment does not extend outside of the TCP segment boundary.

26. The NIC of claim 21, wherein the conducting means conducts c), 2) in the case that the data transfer includes a single DDP segment and an MPA length associated with the single DDP segment is greater than a transmission control protocol (TCP) segment length of the data transfer.

27. The NIC of claim 21, wherein the conducting means conducts c), 3) in the case that the data transfer includes a single DDP segment that has: an MPA length associated with the single DDP segment that equals a TCP segment length, and a valid CRC.

28. The NIC of claim 21, wherein the conducting means conducts c), 1) in the case that the data transfer includes a single DDP segment that has: an MPA length associated therewith that equals a TCP segment length, an invalid CRC and has a DDP header that is referred by an MPA length associated with a previous DDP segment.

29. The NIC of claim 28, wherein in the case that the single DDP segment that has: an MPA length associated therewith that equals a TCP segment length, an invalid CRC, and a DDP header that is not referred by an MPA marker, the conducting means conducts:

c), 1) in the case that the DDP header is referred by an MPA marker; and

c), 2) in the case that the DDP header is not referred by an MPA marker.

30. The NIC of claim 19, further comprising means for setting the data transfer type to the first type when the conducting means conducts c), 2).

31. The NIC of claim 19, further comprising means for clearing TCP hole information in a connection context and stopping receipt reporting for an out-of-order data transfer upon which the means for conducting conducts c), 3).

32. The NIC of claim 19, wherein the data transfer includes DDP segments, and the calculating means calculates a CRC for all DDP segments of the data transfer together.

33. The NIC of claim 19, wherein the data transfer does not contain an MPA marker.

34. The NIC of claim 19, further comprising:

third means for storing a number of retransmission attempts for each data transfer including an error; and

fourth means for storing a largest sequence number.

35. The NIC of claim 34, wherein in the case that CRC is invalid for the data transfer, which indicates the data transfer is a newly received error-including data transfer:

the conducting means conducts c), 2) on the newly received error-including data transfer in the case that the number of retransmission attempts exceeds a maximum retransmission attempt number for that data transfer, and

the conducting means conducts c), 1) on the newly received error-including data transfer in the case that the number of retransmission attempts does not exceed a maximum retransmission attempt number for that data transfer; and

the NIC further comprising:

means for increasing the number of retransmission attempts for the newly received error-including data transfer by one in the case that the conducting means conducts c), 1); and

means for updating the fourth storing means to carry the largest sequence number among at least one previously received error-including data transfer and the newly received error-including data transfer in the case that the conducting means conducts c), 1).

36. The NIC of claim 34, further comprising:

means for resetting the number of retransmission attempts in the case that the CRC is valid for an in-order data transfer, and a sequence number of the in-order data transfer is greater than the stored largest sequence number; and

wherein the conducting means conducts c), 3) in the case that:

a) the CRC is valid for an in-order data transfer and the sequence number of the in-order data transfer is not greater than the stored largest sequence number, and

b) the resetting means resets the number of retransmission attempts.

37. A computer program product comprising a computer useable medium having computer readable program code embodied therein for handling a data transfer in a network interface controller (NIC), the program product comprising the steps of:

program code configured to receive the data transfer wherein the data transfer is denoted as one of a first type and a second type;

program code configured to calculate a cyclical redundancy check (CRC) for the data transfer, wherein the CRC is one of valid and invalid;

program code configured to conduct one of:

1) dropping the data transfer and not confirming reception;

2) placing the data transfer to a reassembly buffer of the NIC; and

3) placing the data transfer to an internal buffer of the NIC for direct data placement to a destination buffer.

38. The program product of claim 37, further comprising program code configured to set the data transfer type to the first type when the conducting program code conducts c), 2).

39. The program product of claim 37, further comprising program code configured to clear TCP hole information in a connection context and stop receipt reporting for an out-of-order data transfer upon which the conducting program code conducts c), 3).

40. The program product of claim 37, wherein the conducting program code conducts c), 2) in the case that the data transfer is of the first type.